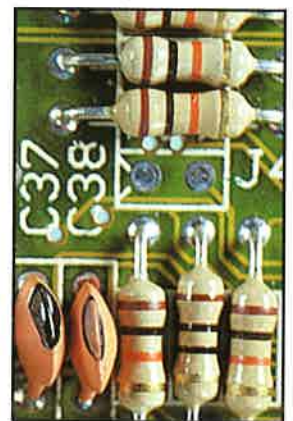
Il 386

Il microprocessore 386 è stato commercializzato verso la fine del 1985 ed ha rappresentato un passo importantissimo nello sviluppo dei personal computer; infatti, ha permesso di superare la barriera dei 16 bit dei microprocessori precedenti (8086 e 80286), consentendo di gestire dei bus da 32 bit.

Il vantaggio principale che questo microprocessore presenta rispetto ai precedenti è costituito dalla sua capacità di elaborare dati composti da parole di 32 bit, invece dei 16 bit tradizionali del 286, consentendo una velocità di elaborazione decisamente superiore.

Anche l'architettura interna è stata sostanzialmente migliorata, raddoppiando il numero dei transistor integrati: 275.000 rispetto ai 134.000 dell'80286. L'implementazione dell'architettura interna è stata sviluppata non dimenticando la compatibilità di questo microprocessore con i precedenti 8088, 8086 e 80286; infatti, questo processore è in grado di gestire anche i software scritti per i micro precedenti.

Il microprocessore 386 può elaborare dati in parole da 32 bit

Una delle caratteristiche più importanti del 386 è costituita dalla memoria cache

In effetti la compatibilità del 386 non è frutto del caso, perché il 386 è stato sviluppato con queste specifiche proprio per poter ancora utilizzare le migliaia di applicazioni software scritte per i microprocessori a 16 bit.
Un'altra caratteristica rilevante, che lo rende superiore ai suoi predecessori, è costituita dall'aumentata capacità di memoria che è in grado di gestire: può infatti indirizzare sino a 4 Gbyte (2 elevato a 32 indirizzi). Inoltre, ha aperto una nuova dimensione al PC grazie alla capacità di supportare diverse operazioni contemporanee in modo non simulato e alla possibilità di operare in modo ottimale in ambiente UNIX.
La sua velocità di elaborazione è decisamente superiore a quella dei microprocessori precedenti, quali il 286, poiché è in grado di gestire tre o quattro milioni di istruzioni al secondo.

Il 386 sfonda la barriera dei 16 bit, perciò è in grado di elaborare i dati più rapidamente rispetto ai microprocessori precedenti (80286, 8086)

L'AUTODIAGNOSI

Il 386 è dotato di un sistema automatico di autotest che viene eseguito ad ogni avvio dell'elaboratore. L'autodiagnosi prevede il controllo delle sue principali funzioni interne, quali il buon funzionamento della memoria ROM e della logica del microprocessore.
Questa operazione di controllo viene eseguita in un periodo di tempo molto breve di circa 30 ms, e il risultato viene memorizzato in due registri destinati allo scopo.

LA MEMORIA CACHE

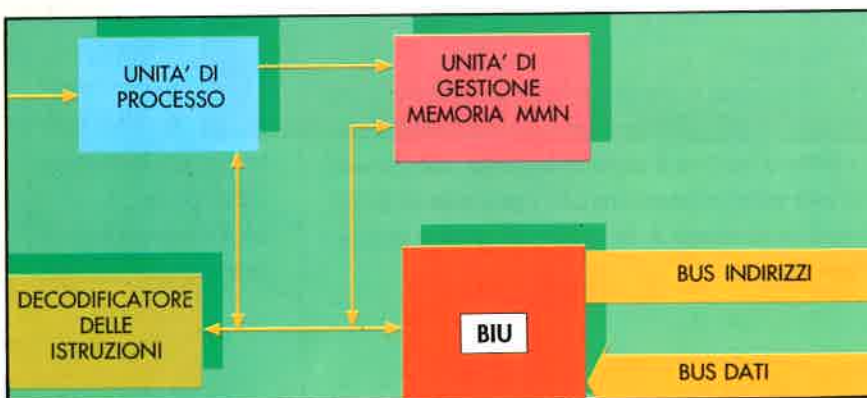
Una caratteristica molto importante del 386 è costituita dal fatto che una certa quantità di memoria, conosciuta con il nome di "CACHE", viene riservata per memorizzare gli indirizzi di memoria più utilizzati; in questo modo si ottiene una diminuzione dei tempi di elaborazione poiché l'accesso a questa memoria è praticamente istantaneo.
Questi indirizzi sono i puntatori della memoria esterna, da dove vengono prelevati i dati richiesti.

L'INTERNO DEL 386

Il 386 è realizzato con una struttura di tipo *a cascata*, anche conosciuta con il nome di "pipeline" o "segmentata", che consente di aumentarne la velocità di elaborazione.
Questo tipo di struttura prevede la suddivisione del processo complessivo di elaborazione in diverse fasi o stadi perfettamente definiti:

1. - ricerca dell'istruzione (RI)
2. - decodificazione (D)
3. - ricerca degli operandi (RO)
4. - esecuzione (E).

Il 386, eseguendo sequenzialmente le istruzioni previste dalle suddette fasi al ritmo imposto dagli impulsi generati dal clock, riduce il tempo di elaborazione, e di conseguenza la velocità risulta decisamente più elevata.
Internamente è costituito da tre grandi moduli, che sono i seguenti:

Schema a blocchi caratteristico di un microprocessore 386

- *CPU "Unità Centrale di Elaborazione"*
- *MMU "Unità per la Gestione della Memoria"*
- *BIU "Unità di Interconnessione con il Bus".*

La **CPU** ha il compito di controllare e gestire il microprocessore, ed è composta da quattro parti nettamente distinte:

- *blocco dei registri*
- *ALU, "Unità Aritmetico Logica"*
- *modulo per la decodifica delle istruzioni*
- *blocco di ricerca dei codici.*

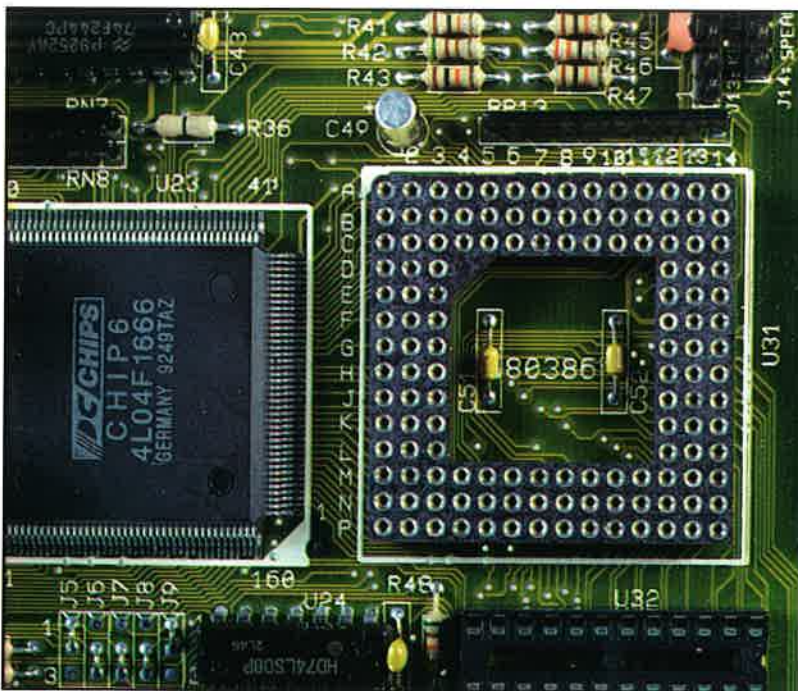
La **MMU** esegue la manipolazione e la gestione della memoria, sia in modalità paginata che in quella segmentata. Quando si utilizza la memoria segmentata è possibile proteggere, se l'utente lo desidera, alcune applicazioni.

É inoltre possibile operare con la memoria nelle due modalità già descritte anche per il 286, quella reale e quella protetta.

Struttura interna che rappresenta i diversi elementi di un microprocessore 386

In modalità reale l'indirizzamento, la dimensione della memoria e l'utilizzo delle istruzioni rende il suo funzionamento simile a quello di un 286, con l'unica differenza che il 386 opera con registri a 32 bit.

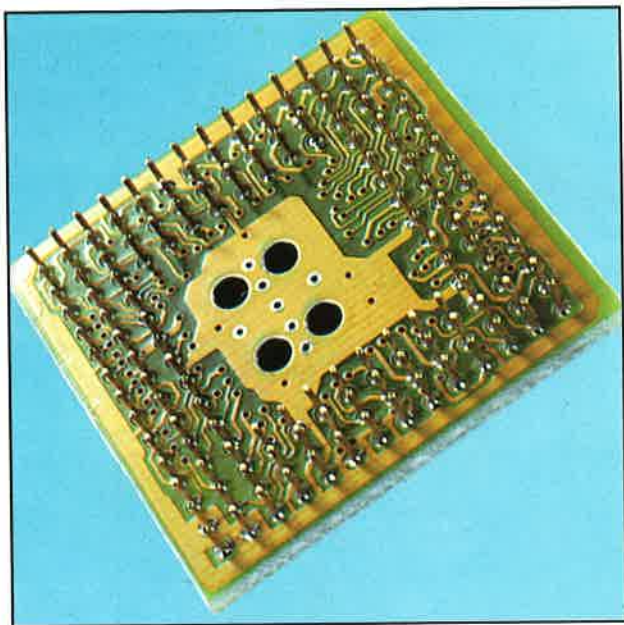
In modalità protetta si possono invece indirizzare fino a 4 Gbyte di memoria RAM, il che implica possibilità di lavoro decisamente superiori, come ad esempio la multiutenza o il multitasking in ambiente UNIX.

Il 386 deve essere inserito in uno zoccolo molto affidabile

La **BIU** è l'unità che controlla l'interscambio delle informazioni con il mondo esterno attraverso i bus, e le sue funzioni possono essere richieste:

- dall'unità di ricerca quando fornisce l'indirizzo dell'istruzione successiva da memorizzare nella coda di prericerca,
- dall'unità di esecuzione, quando preleva gli operandi dalla memoria o dalle interfacce di I/O, o quando restituisce i risultati,
- dalla MMU, quando fornisce gli indirizzi relativi ai dati da elaborare.

Il funzionamento della BIU è decisamente ottimizzato poiché, quando non sta eseguendo delle istruzioni, si dedica alla ricerca del codice relativo

In modalità protetta il 386 può indirizzare fino a 4 Gbyte di memoria RAM

Il 386 è dotato di 36 registri, la metà dei quali può essere utilizzata per i programmi applicativi

Vista della faccia inferiore del 386, nella quale si può notare l'elevato numero di terminali

all'istruzione successiva e lo memorizza temporaneamente in una coda da 16 byte.

I PROGRAMMATORI DEL 386

Per lo sviluppo del software dedicato al 386 sono necessarie due categorie di programmatori: i "programmatori per gli applicativi" e i "programmatori di sistema".

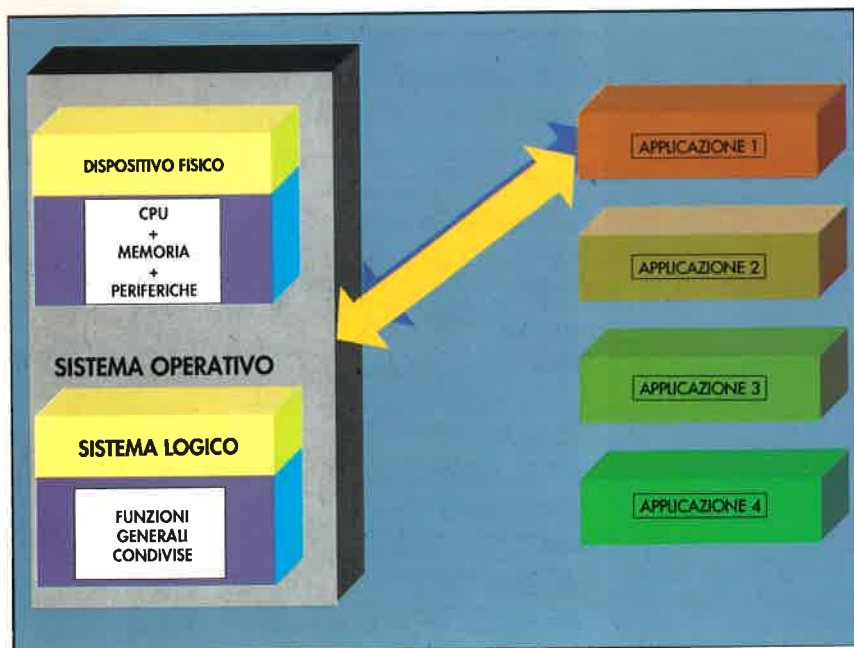
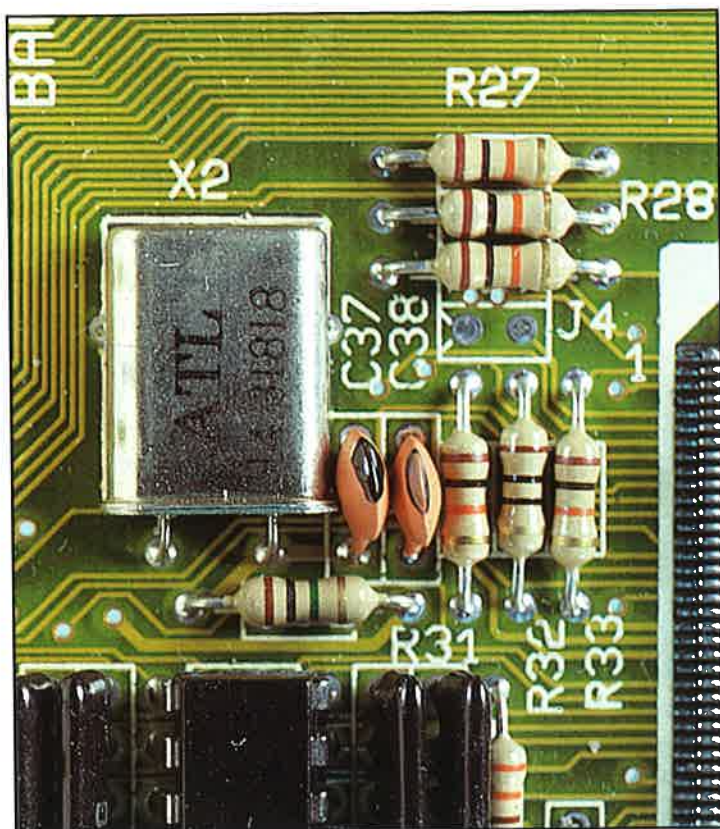
Il 386 supporta la multiutenza, che può essere gestita per mezzo di un opportuno sistema operativo in grado di amministrare i tempi di esecuzione delle operazioni che vengono richieste in modo contemporaneo al microprocessore

Il *programmatore per gli applicativi* ha il compito di realizzare un sistema logico che supporti le applicazioni dell'utente. Questo tipo di programmazione guarda alla CPU come ad un insieme di registri di lavoro che consentono di gestire istruzioni, dati, indirizzi e altri elementi di memoria. Il *programmatore di sistema* deve invece realizzare un software che permetta lo sfruttamento ottimale dell'elaboratore, in modo da renderlo in grado di supportare tutte le applicazioni previste alla massima potenzialità e con la massima sicurezza.

I REGISTRI

Questo microprocessore è dotato di 36 registri, dei quali la metà sono a disposizione dei programmatori per i software applicativi. Tra i più importanti si possono ricordare i seguenti:
- *registri di uso generale*
- *registro del contatore di programma*
- *registro degli indicatori*
- *registro dei segmenti.*

I registri di uso generale servono per memorizzare dati che vengono utilizzati in una fase successiva, e sono costituiti da otto registri in grado di gestire informazioni da 32 bit quando utilizzano tutta la loro dimensione.
Possono però essere impiegati come registri da 16 bit, e quattro di questi da 8 bit.

Il contatore di programma (EIP) è un registro a 32 bit che memorizza lo scorrimento che bisogna applicare alla base del segmento del codice per poter localizzare il punto in cui è situato l'indirizzo dell'istruzione che deve essere eseguita successivamente.
Il registro degli indicatori viene anche chiamato EFLAGS, ed è costituito da 32 bit dei quali una parte sono destinati agli indicatori di stato controllati dalla ALU, quali il riporto, la parità, il riporto ausiliario, lo zero, il segno e l'overflow. I bit rimanenti agiscono come indicatori di sistema legati al meccanismo di protezione ed ad altri mezzi di cui è dotata la CPU.
Un importante gruppo di registri sono quelli di segmento, utilizzati per memorizzare gli indirizzi relativi ai segmenti che si stanno utilizzando in un determinato istante di una determinata operazione.
Esiste anche un gruppo di registri che vengono

Oscillatore al quarzo necessario per fornire la frequenza di clock al microprocessore

impiegati per il collegamento opzionale del coprocessore matematico, con il compito di gestire la programmazione degli applicativi e il set delle istruzioni.

Struttura della memoria

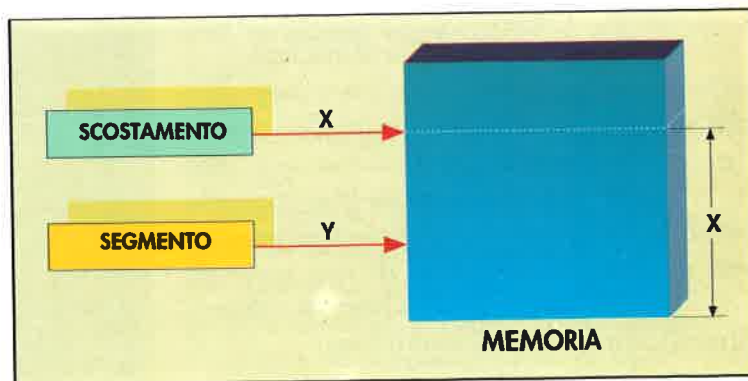
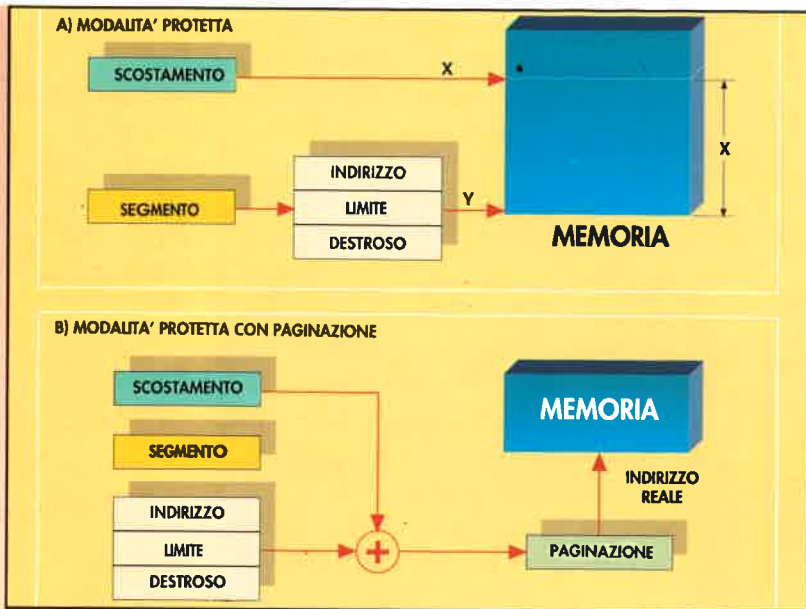
La memoria gestita dal microprocessore 386 è composta da byte, parole e parole doppie.
La *parola* è formata da due byte, mentre la *parola doppia* da quattro byte strutturati in ordine di peso.
Ciò significa che quello di minor peso o meno significativo corrisponde all'indirizzo più basso, e quello di maggior peso o più significativo all'indirizzo più alto.
La gestione della memoria avviene tramite i segnali BE0#-BE3#, con i quali è possibile selezionare uno qualsiasi dei quattro byte relativi a una posizione di memoria da 32 bit.
Il 386 è in grado di gestire anche altre strutture di dati molto più complesse, quali i segmenti e le pagine.
I *segmenti* sono blocchi di memoria che contengono informazioni di uguali caratteristiche, e sono utili per la programmazione strutturata mediante l'organizzazione della memoria in moduli logici dello stesso tipo.
La *pagina* consiste in una suddivisione dello spazio di memoria in settori con dimensione fissa di 4 Kbyte. Con le pagine si semplificano gli algoritmi di interscambio tra oggetti della memoria fisica e della memoria virtuale, con conseguente aumento della velocità e riduzione del lavoro per i programmatori di sistema.
Queste due strutture di dati non si escludono a vicenda, ma sono tra di loro complementari; il 386 è infatti in grado di gestire contemporaneamente le due tecniche sfruttando i principali vantaggi di entrambe.
I sistemi operativi DOS e UNIX operano con queste tecniche di strutturazione della memoria.

La memoria gestita dal microprocessore 386 è composta da byte, parole, e doppie parole di indirizzo

Gestione della memoria

Il 386 ha due modalità di funzionamento:
- *modalità reale,*
- *modalità protetta.*

In modalità reale un indirizzo viene gestito sommando al registro di partenza del segmento delle istruzioni lo scostamento contenuto nel registro puntatore degli indirizzi. In questo modo la capacità massima di indirizzamento della memoria è di 1.024 Kbyte, e l'unica differenza rispetto ai microprocessori precedenti (8086, 80286) è dovuta alla capacità del 386 di gestire registri a

Schema di indirizzamento della memoria in modalità reale

Schema di indirizzamento in modalità protetta, senza paginazione e con paginazione

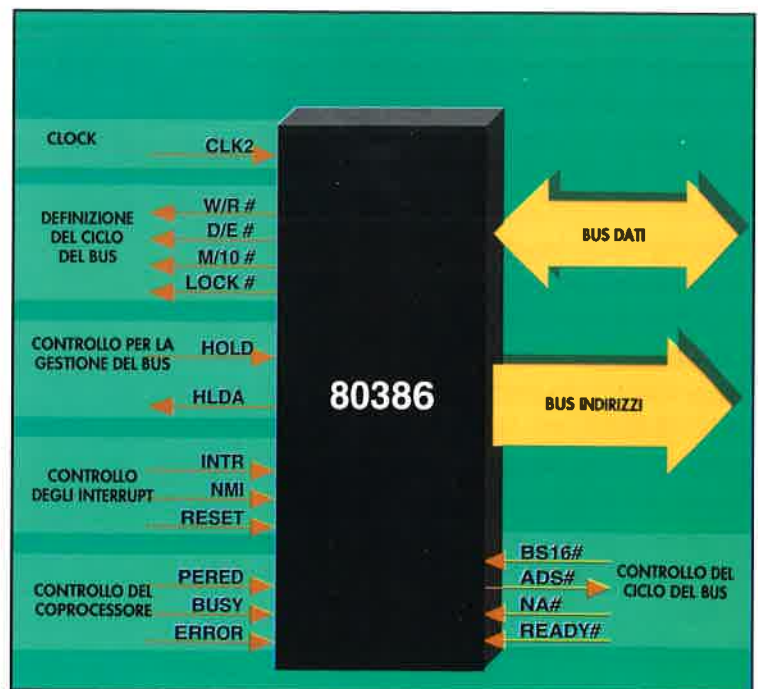
32 bit. In questa modalità di lavoro non è possibile utilizzare la paginazione, poiché gli indirizzi virtuali coincidono con quelli fisici.
In modalità protetta il 386 può indirizzare sino a 4 Gbyte di memoria, suddivisa in segmenti che possono avere una dimensione compresa tra 1 byte e 4 Gbyte.
Ogni segmento è a sua volta suddiviso in una o più pagine, e il sistema con cui viene individuato l'indirizzo di memoria è il seguente: l'indirizzo di base è contenuto in una tabella ed è composto da 32 bit. L'indirizzo reale si ottiene aggiungendo all'indirizzo di base prelevato dalla tabella un determinato scostamento. L'accesso alla tabella degli indirizzi di base avviene tramite un selettore.
Se si utilizza la paginazione il processo è più complesso, poiché esistono alcune tabelle di memoria che contengono tre tipi di descrittori: locali, globali e di interrupt.
I descrittori locali sono quelli riferiti ad una determinata operazione, e contengono l'informazione relativa all'operazione di un segmento.
I descrittori globali contengono una informazione di tipo generale, mentre quelli di interrupt indicano i vettori di interruzione.

Una pagina di memoria consiste nella suddivisione dello spazio occupato dalla memoria in settori di una certa dimensione

Il sistema con il quale il 386 genera l'indirizzo reale è il seguente: prima calcola l'indirizzo della pagina in cui si trova, sommando l'indirizzo di ingresso della tabella degli indirizzi con l'indirizzo di partenza della stessa. Successivamente l'indirizzo di questa pagina viene sommato all'indirizzo di ingresso della tabella delle pagine, e il risultato di questa somma viene sommato allo scostamento, ottenendo così l'indirizzo reale di memoria in cui si trova il dato.
Questo tipo di indirizzamento è stato sviluppato per poter operare in multiutenza, gestita da un sistema operativo che la supporta, come può essere l'UNIX.
Il 386 è dotato anche di alcuni sistemi di protezione interni che servono ad evitare che qualche programma interferisca con altri, e che non si verifichino perdite di dati a causa della loro interazione.

Gli interrupt e le eccezioni

Gli *interrupt* e le eccezioni sono operazioni che alterano la normale elaborazione delle istruzioni del programma in corso; in queste situazioni viene richiesta al microprocessore l'esecuzione di routine specifiche che riguardano gli avvenimenti esterni o interni che le hanno provocate.
Gli *interrupt* sono provocati da avvenimenti ester-

Segnali di I/O del 386 suddivisi per gruppi funzionali

ni che abilitano un segnale di richiesta inviato al microprocessore attraverso alcuni terminali dedicati allo scopo.
Le *eccezioni* si generano automaticamente come conseguenza di una condizione anormale avvenuta e rilevata durante lo svolgimento del programma in esecuzione.
Gli interrupt possono essere così classificati:
- *interrupt esterni,*
- *Interrupt interni.*

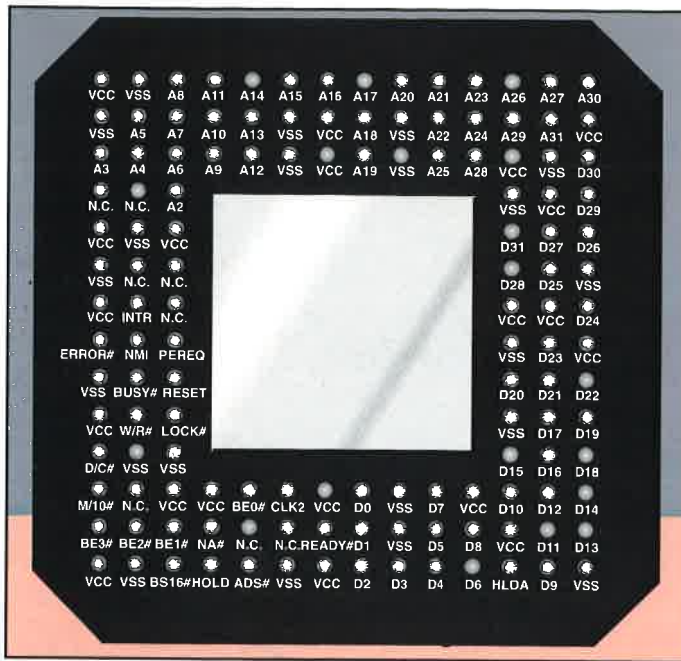
Gli interrupt esterni sono quelli provocati dall'hardware, e sono generati da avvenimenti asincroni esterni che si materializzano come conseguenza dell'attivazione di segnali applicati ai terminali NMI e INTR.
L'interrupt NMI (interrupt non mascherabile) si caratterizza poiché è sempre vigile, in attesa di avvenimenti esterni catastrofici o di massima urgenza (come possono essere la mancanza dell'alimentazione, errori nella memoria, ecc.).

La periferia del 386 è costituita da un hardware opportuno, necessario per il suo funzionamento

L'interrupt INTR gestisce un circuito di controllo, come ad esempio l'8259, che assegna una priorità agli otto possibili richiedenti che supporta associando a ciascuno di essi un vettore di interrupt corrispondente.
Per interrupt interni si intendono quelli originati dal software e che si materializzano come conseguenza dell'esecuzione, in determinate condizioni, di alcune istruzioni speciali; questi interrupt sono l'INT e l'INTO.
Con l'INT si può verificare in qualsiasi parte del programma il comportamento della routine di interrupt che si desidera, mentre con l'INTO viene generata una richiesta di interruzione che il processore soddisfa utilizzando l'ingresso 4 della tabella.
Le eccezioni sono generate automaticamente dal 386 quando rileva un errore o una situazione speciale, e si possono dividere in tre categorie:
- *guasti o errori*, sono eccezioni che vengono rilevate e corrette prima di eseguire l'istruzione che ha generato l'errore; un classico esempio si verifica quando il microprocessore effettua una chiamata ad un segmento o a una pagina che non si trova nella memoria. In questo caso la routine di attenzione del sistema operativo ricerca il segmento o la pagina nella memoria virtuale e la carica nella memoria principale; a partire da questo punto il 386 riprende l'esecuzione dell'istruzione che ha generato l'eccezione.
- *trappole*, anche chiamate semplicemente eccezioni; sono gli interrupt definiti dall'utente e compresi nel programma.
- *aborti*, eccezioni che consentono la localizzazione esatta dell'istruzione che ha originato la situazione anomala.

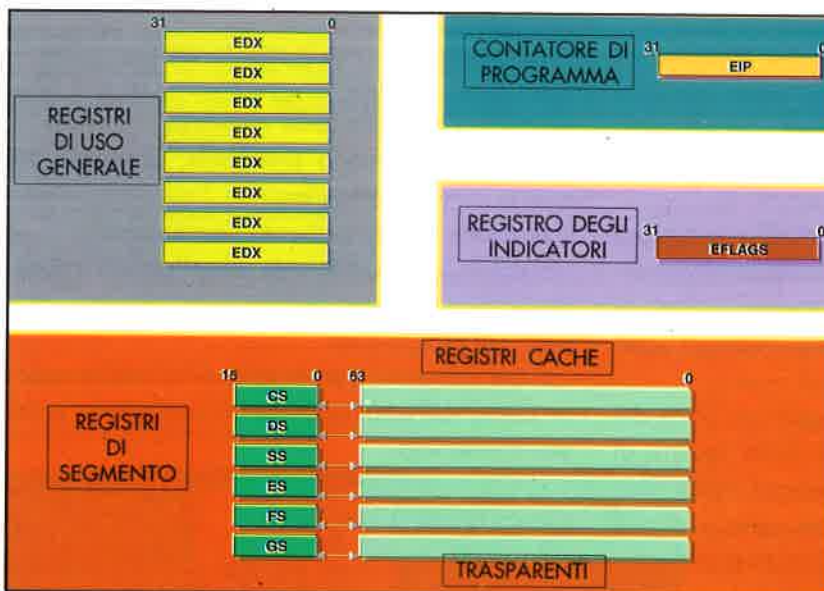
Connessioni del 386

Il 386 supporta una serie di segnali sui diversi terminali che possono essere classificati nei seguenti gruppi:
- *ALIMENTAZIONI*, corrispondenti ai terminali Vcc (20) collegati alla tensione continua di 5 Vcc, e ai terminali Vss (21) collegati a massa,
- *SEGNALI DI CLOCK*, corrispondente al terminale

Gli interrupt sono generati da avvenimenti esterni che abilitano un segnale inviato a determinati terminali del microprocessore

Distribuzione dei terminali di un 386

CLK2 che riceve il segnale di clock necessario per il controllo interno delle sincronizzazioni; questa frequenza viene divisa internamente per due in modo da generare la frequenza di lavoro del microprocessore; di conseguenza, un 386 a 33 MHz necessita di un segnale di clock da 66 MHz,

- *BUS DEI DATI*, composto da 32 terminali di tipo bidirezionale e tristate, indicati con le sigle da D0 a D31, incaricati del trasferimento dell'informazione,
- *BUS DEGLI INDIRIZZI*, composto da 32 terminali di uscita di tipo tristate, indicati da A0 a A31, che gestiscono la memoria e i dispositivi di ingresso e uscita,
- *SEGNALI DI STATO DEL BUS*, sono 5 linee di uscita di tipo tristate, incaricate di definire lo stato del bus attivo; questi terminali sono: ADS#, W/R, D/C, M/10#, LOCK.
- *SEGNALI DI CONTROLLO DEL BUS*, controllano la realizzazione delle operazioni compiute sul bus attivo (READY), la larghezza del bus dati (BS16#), e la possibilità di ottenere l'indirizzo da utilizzare in un ciclo prima che venga completato il ciclo precedente (NA#).

Gli interrupt esterni sono quelli provocati dall'hardware

Registri disponibili per i programmi applicativi

I CONTATORI ASINCRONI

Una delle applicazioni tipiche dei registri di scorrimento esaminati nel capitolo precedente è costituita dai contatori. Questi possono essere suddivisi in due grandi categorie: contatori asincroni e contatori sincroni.

Un contatore può essere considerato come un circuito in grado di ricordare il numero degli impulsi che gli sono stati applicati.

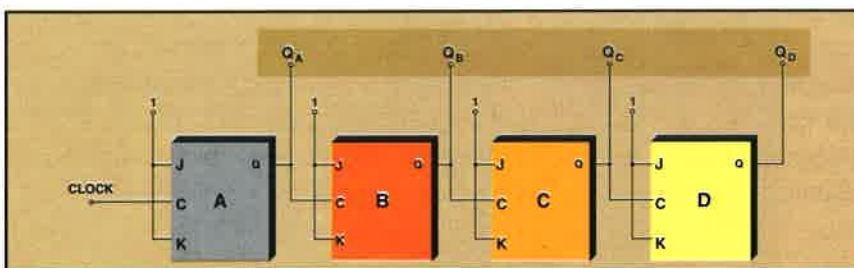
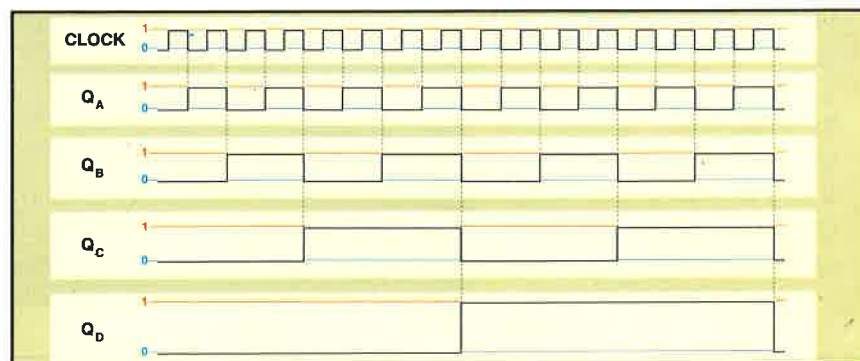
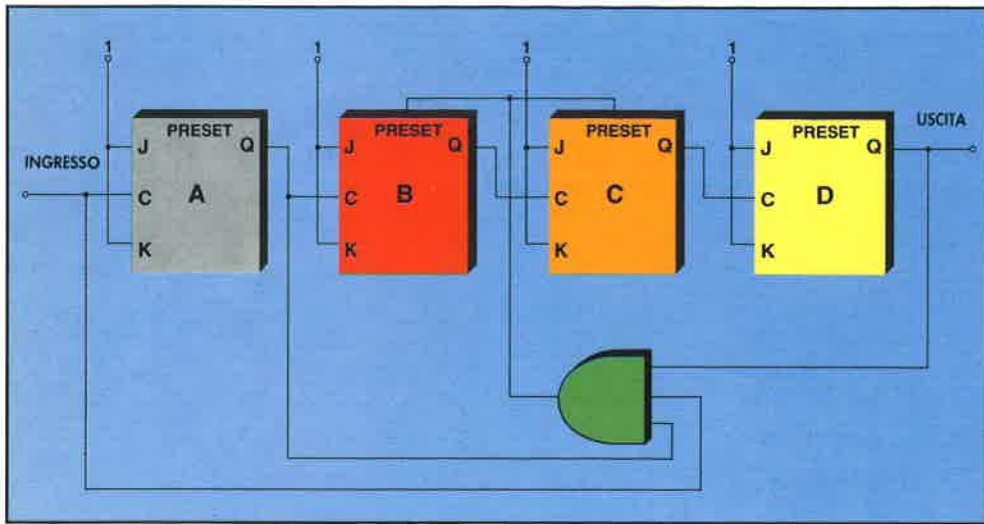
Essenzialmente, i contatori digitali sono catene di bistabili reazionati con diversi sistemi. La loro importanza è notevole per le numerosissime applicazioni a cui sono destinati, tra le quali si segnalano: acquisizione di dati, divisione di frequenza, ecc. Prima di iniziare lo studio dei contatori asincroni, è opportuno esaminarne il principio di funzionamento con riferimento a dispositivi particolari definiti *contatori ad anello*. Se si collega l'uscita del bit meno significativo di uno shift register a cinque bit all'ingresso serie, si ottiene una memoria a scorrimento che può essere definita un contatore ad anello, o chiuso. Si supponga che tutti i flip-flop si trovino a 0 e che l'ultimo sia abilitato in modo tale che la sua uscita sia a 1 mentre quelle degli altri flip-flop sono a 0. Il primo impulso di clock trasferisce lo stato dell'ultimo bistabile al primo della catena, per cui l'uscita di questo diventa 1 mentre le restanti rimangono o si portano a 0. Gli impulsi di clock successivi trasfe-

Con una catena di quattro bistabili di tipo T è possibile ottenere un dispositivo definito contatore asincrono

Diagramma temporale delle uscite di ogni bistabile in funzione del segnale di clock

Esistono alcune varianti dei contatori asincroni, come ad esempio questo contatore decadico

Aggiungendo due circuiti AND-OR si ottiene un contatore asincrono reversibile

riscono progressivamente lo stato dell'uscita precedente a quella successiva secondo un percorso chiuso che viene chiamato anello. Il conteggio degli impulsi di clock applicati può essere "letto" semplicemente osservando quale flip-flop si trova alla stato 1, e per fare questo non è necessario utilizzare un decodificatore.
Questi contatori ad anello non sfruttano completamente i flip-flop, poiché con cinque stadi si ottiene un contatore con rapporto 5:1, mentre 5 flip-flop possono definire 32 stati (ottenuti elevando 2 alla quinta potenza).

CONTATORI ASINCRONI FONDAMENTALI

Viene presa in esame una catena di 4 flip-flop J-K master-slave, ognuno con l'uscita collegata all'ingresso di clock del bistabile seguente, come si può osservare nella figura corrispondente.
Gli impulsi che devono essere conteggiati vengono applicati all'ingresso di clock del flip-flop A. In tutti gli stadi gli ingressi J e K sono collegati all'alimentazione, in modo che il loro valore logico sia sempre 1. Questa condizione converte ogni stadio in un bistabile di tipo T.
Bisogna ricordare che per un flip-flop di tipo T con ingresso a 1 il master cambia di stato ogni volta che il segnale di clock passa da 0 a 1, e che la nuova uscita del bistabile viene trasferita allo slave quando il clock passa da 1 a 0. Questo cambio di stato richiede le seguenti condizioni:
- che l'uscita Q_A cambi di stato durante il fronte di discesa di ciascun impulso,
- che tutte le altre uscite eseguano una transizione quando l'uscita del flip-flop precedente passa da 1 a 0. Questa condizione è valida per tutti i bistabili che formano il contatore, dal bit meno significativo al bit più significativo.

Seguendo queste due regole si ottiene il diagramma temporale rappresentato nella figura corrispondente. La tabella che segue riporta lo stato di tutti i bistabili in funzione del numero di impulsi esterni applicati:

Q_D	Q_C	Q_B	Q_A	N. IMPULSO
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Questa tabella può essere verificata direttamente con le forme d'onda del diagramma corrispondente. Come si può facilmente osservare, in ogni linea le uscite attive corrispondono alla rappresentazione binaria del numero degli impulsi di ingresso, per cui la catena dei bistabili esegue un conteggio nel sistema binario. Tra i contatori asincroni vi sono alcune varianti che è opportuno esaminare.

Con una semplice porta NAND, e abilitando il segnale asincrono di Clear, si può realizzare un contatore decadico

CONTATORE ASINCRONO DECADICO

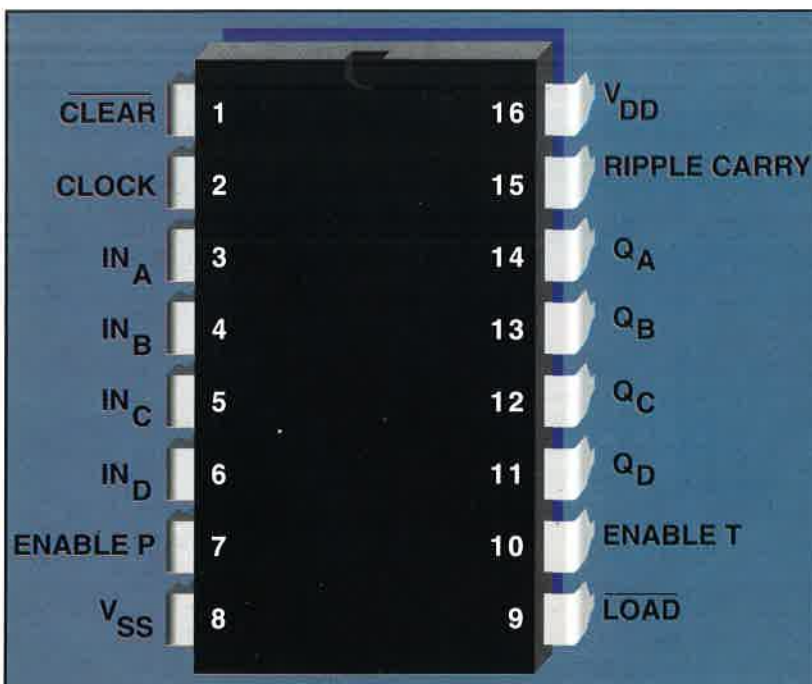
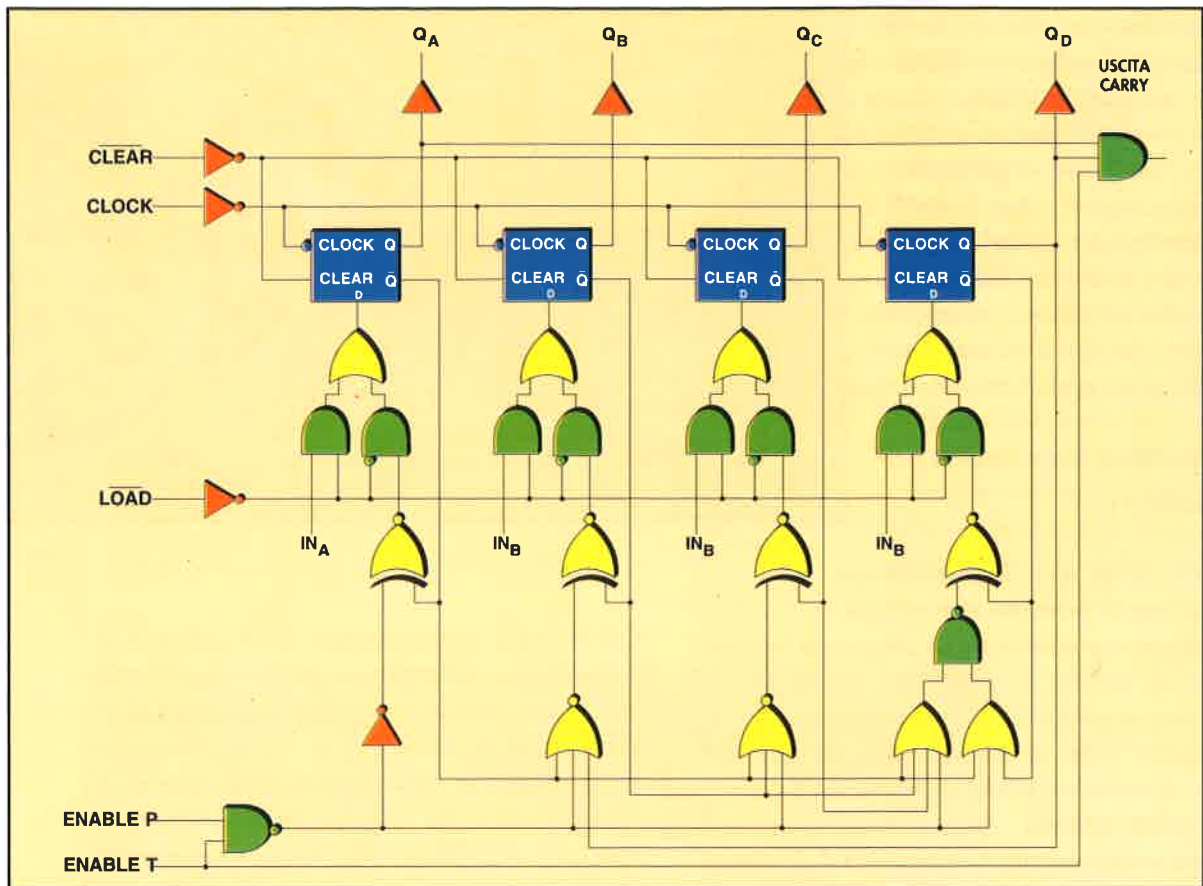
Si tratta di un contatore asincrono particolare di notevole importanza. Essenzialmente costituisce un divisore di frequenza per un numero di impulsi pari a 10. La sua struttura può essere facilmente ricavata dalle considerazioni fatte in precedenza, ed è quindi interessante vederne il funzionamento. Il numero dei flip-flop richiesti corrisponde al valore che bisogna assegnare alla potenza di due per ottenere almeno il numero degli impulsi applicati (che in questo caso è pari a 10); l'esponente necessario sarà 4, poiché $2^3 = 8 < 10$, per cui non si ottiene il numero di impulsi applicati, e $2^4 = 16 > 10$. In accordo con questa premessa, il divisore è costituito dal circuito indicato nella rispettiva figura.

Il circuito esegue un conteggio con il codice BCD (Binary Code Decimal), nel quale i digit binari hanno come peso le prime quattro potenze di 2, vale a dire 8, 4, 2, 1.

I nove primi stati del contatore sono identici a quelli del contatore binario. La transizione, o ritorno allo stato 0 dallo stato 9, avviene nel modo seguente:

1. - sul fronte di salita dell'impulso di ingresso numero 10 tutti i flip-flop si portano allo stato 1 a causa della reazione sul segnale di Preset dei bistabili B e C,

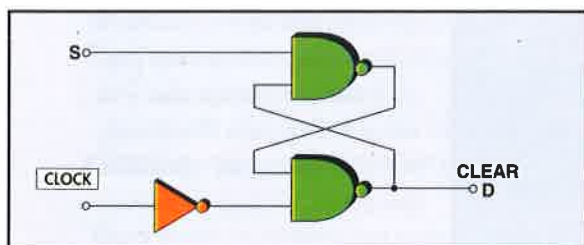
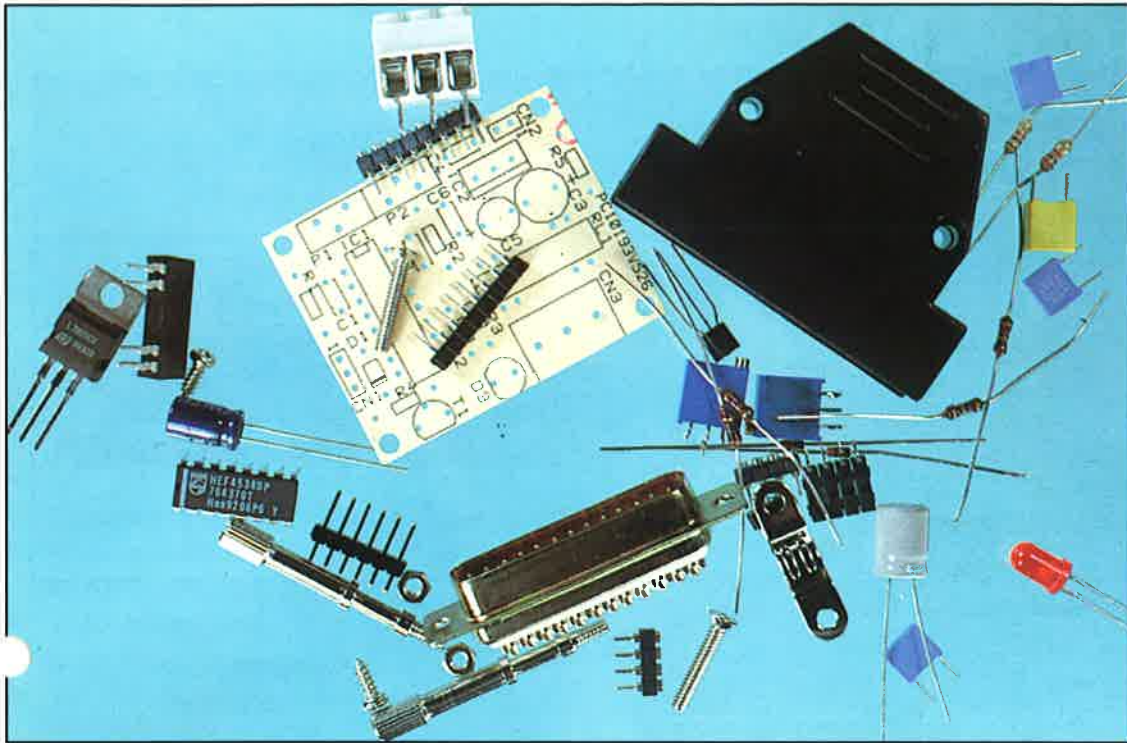
2. - sul fronte di discesa dell'impulso di ingresso numero 10 tutti i flip-flop cambiano di stato e sulle loro uscite sarà presente uno 0.

Da quanto visto sinora relativamente ai contatori asincroni, già si possono osservare le limitazioni che li rendono inservibili per determinate applicazioni. A volte infatti, il contatore passa transitoriamente per una serie di stati non desiderati. Ad esempio, nel contatore decadico si può osservare

Disposizione dei terminali di un circuito integrato che contiene un contatore decadico controllato dal segnale asincrono di Clear

Elementi circuitali interni di un contatore decadico integrato

che per effetto dei ritardi nei bistabili si verificano degli stati di transizione quando viene eseguito il salto da 9 a 10.

La separazione tra gli impulsi oggetto del conteggio, corrispondente alla massima frequenza di lavoro del contatore, dipende dalla somma dei ritardi di ciascun bistabile; questi ritardi sono cumulativi poiché ciascun flip-flop viene innescato dalla transizione dell'uscita che lo precede.

Gli stati stabili del contatore non vengono sempre raggiunti in modo contemporaneo, ma dipende dal numero dei flip-flop che devono innescarsi per raggiungere lo stato previsto.

Tutti questi inconvenienti si risolvono in parte o totalmente con il contatore sincrono, caratterizzato soprattutto dal fatto che i bistabili di cui è composto si innescano contemporaneamente. Questa condizione si ottiene aggiungendo qualche elemento al circuito combinatorio di reazione dei flip-flop, che nel caso di contatori asincroni si riduce quasi sempre ad una sola porta logica.

Per risolvere alcuni problemi legati al ritardo di propagazione è possibile aggiungere questo circuito sull'uscita del contatore decadico

CODIFICATORE MORSE

Quando si deve trasmettere un segnale codificato, che nel caso in esame è un segnale in codice Morse, si possono verificare due situazioni: o si è veramente bravi nella gestione della chiave di codifica o si deve utilizzare il dispositivo che viene presentato di seguito.

Per gli appassionati delle trasmissioni in codice Morse, questo circuito può essere di molto aiuto

In commercio sono già presenti diverse e complesse tecniche elettroniche destinate ad aiutare i radioamatori nella non sempre facile operazione della digitazione di un segnale Morse. Il circuito che viene proposto non vuole essere la panacea della trasmissione automatizzata, ma un semplice e modesto aiuto per tutti coloro che in un modo o nell'altro sono a contatto con entrambi i settori, l'elettronica e le trasmissioni radio.

Schema a blocchi del codificatore Morse. Come si può osservare, il segnale proveniente dal PC viene suddiviso in tre canali

Il vantaggio apparente dell'automatizzazione del processo di scrittura del codice Morse non si manifesta solo quando entra in gioco un personal computer. Da sempre uno dei principali svantaggi della ricezione di un messaggio in codice Morse è dovuto alla relatività della trasmissione rispetto ai diversi operatori (la velocità e la qualità della trasmissione è sempre funzione dell'abilità dell'operatore) poiché, come tutti sanno, nonostante la simbologia Morse sia perfettamente definita ciascun operatore ha un suo stile. Per i neofiti, o meglio ancora per gli hobbisti che non hanno ancora compiuto il passo definitivo per entrare in questo club, il circuito proposto può rappresentare un vero e proprio salvagente.

É doveroso ricordare che lo stesso problema che si vuole risolvere (tramite l'automatizzazione), vale a dire il processo di trasmissione codificata, è già stato risolto da altri. Prova ne è il sistema di trasmissione RTTY (Radioteletype), che ha soppiantato il sistema di trasmissione tradizionale.

Il principale vantaggio dell'RTTY deriva dal fatto che nessun operatore deve digitare il codice da trasmettere

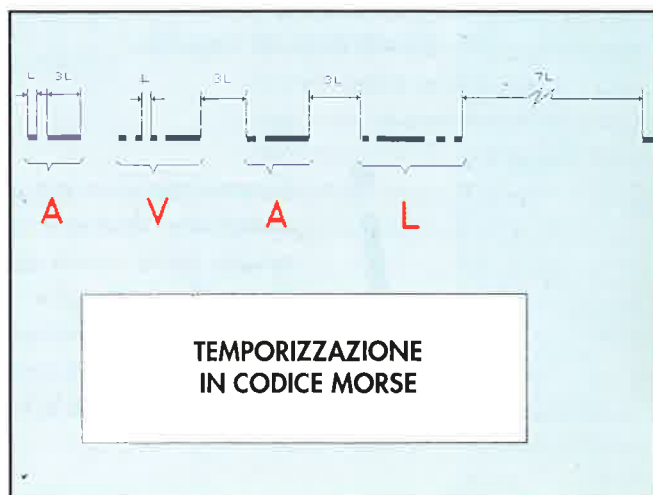
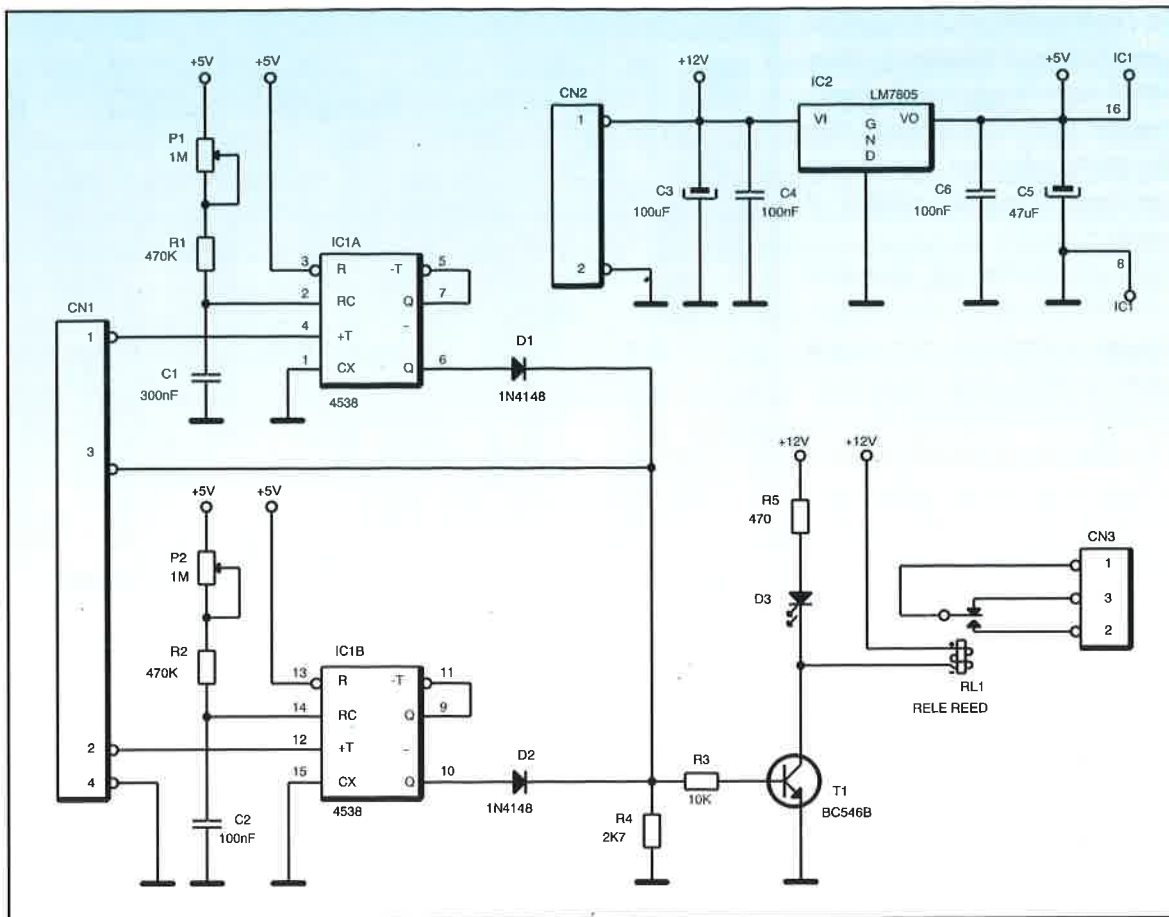
Come si può intuire, il principale vantaggio del sistema RTTY è dovuto al fatto che non è un operatore che scrive il codice che deve essere trasmesso, ma bensì una apparecchiatura complessa che, oltre a semplificare il processo, fornisce un ulteriore possibilità non disprezzabile: una elevata velocità di trasmissione.

Lo scopo che si vuole raggiungere proponendo un circuito di decodifica tramite PC è molto più modesto, anche se grazie al basso costo del dispositivo potrebbero essere molti i radioamatori in possesso di un personal interessati a realizzare il circuito e a conoscerlo nei suoi dettagli.

Il circuito

Come si può osservare dallo schema a blocchi, il segnale proveniente dal PC (in seguito si vedrà come prelevare questo segnale) arriva alla scheda di codifica (che verrà da ora in poi chiamata interfaccia) attraverso il connettore CN1 (dicitura utilizzata nello schema elettronico). Come si può notare il segnale si suddivide in tre canali ben

Normativa delle spaziature tra simboli Morse

Schema elettrico del circuito decodificatore Morse. L'alimentazione richiesta per questo circuito è prelevata da una pila a 9 V, la cui tensione viene stabilizzata a 5 Vcc per mezzo di un circuito regolatore (IC2)

Il cuore del circuito è costituito da un integrato, il CD4538

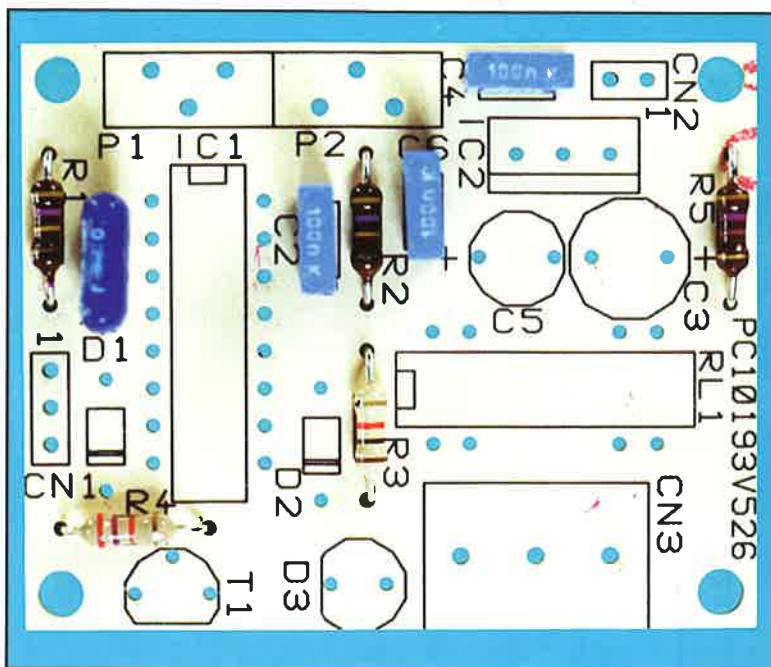
differenziati; i primi due sono di ingresso, e vanno dal PC alla scheda, mentre l'ultimo segue il percorso inverso, dall'interfaccia al PC. I primi due blocchi dell'interfaccia sono costituiti dai generatori di simbolo (in questo caso può essere un punto o una linea). L'uscita di questi due blocchi è collegata ad un circuito che ha il compito di pilotare il relè di uscita, monitorizzare il funzionamento del circuito tramite il diodo LED, e rimandare al PC un segnale che lo informa istantaneamente sullo stato nel quale si trova la trasmissione del simbolo pertinente.
Fanno parte dell'interfaccia anche il relè di uscita, la stabilizzazione dell'alimentazione e tutti gli altri componenti di funzionamento e protezione.
Il collegamento dell'interfaccia all'apparecchiatura di trasmissione dipende in gran parte dalla sua marca e dal modello. Per risolvere questi problemi si è dotato il circuito di un'uscita controllata da un relè, in modo da conferirgli una sufficiente versatilità. Sfruttando il fatto che il relè è dotato sia di un circuito normalmente aperto che di uno normalmente chiuso, si sono collegati tutti i suoi terminali ad un connettore a tre vie (nello schema indicato con CN3), che consente di utilizzare l'ingresso di scrittura dell'apparecchiatura nel modo più opportuno.

FUNZIONAMENTO

Di seguito viene esaminato dettagliatamente il funzionamento del circuito codificatore. Come si può osservare nello schema elettrico, il cuore del circuito è costituito da un circuito integrato, il CD4538.
Questo circuito è internamente dotato di due oscillatori monostabili di precisione; si è deciso di separare la generazione del segnale relativo al punto e del segnale corrispondente alla linea. Lo schema consente di capire questa diversificazione.

Il transistor T1 è il componente incaricato di pilotare il relè di uscita

La parte superiore (nel grafico) del 4538 ha il compito di generare il segnale relativo alla linea, mentre l'altra metà di questo chip (la parte inferiore) lavora per generare il segnale relativo al punto.

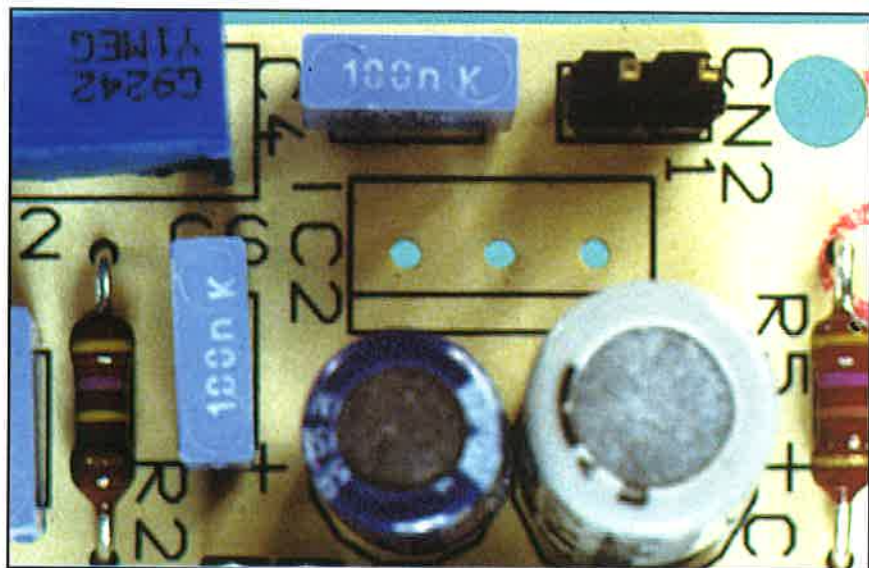
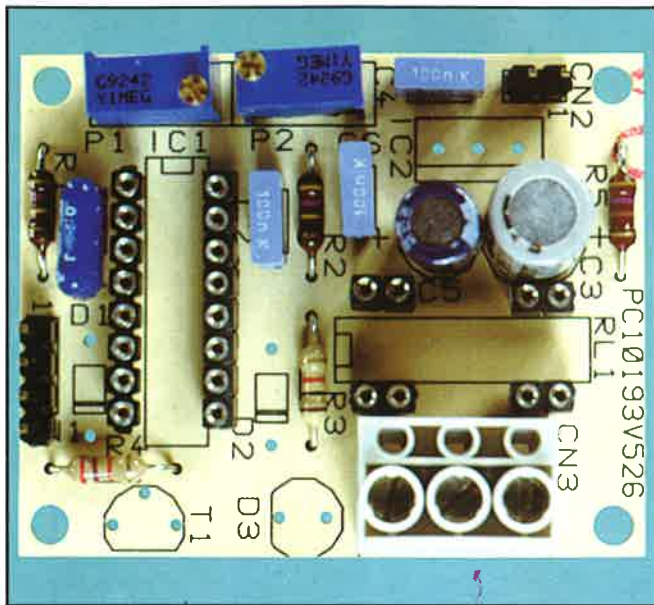
Essenzialmente gli elementi circuitali periferici che accompagnano entrambi i monostabili sono costituiti dagli stessi componenti, ad eccezione di un paio che differiscono tra di loro per il valore nominale (i condensatori C1 e C2). Come è ovvio, questi componenti sono quelli che determinano il tempo di attivazione del monostabile. I valori di questi componenti sono stati scelti in base al tipo di codifica che viene utilizzata. Tuttavia, esiste la possibilità di modificare questi tempi di attivazione dei monostabili (che comporta punti e linee più o meno lunghi); per coloro che vogliono avventurarsi in questi esperimenti, di seguito viene fornita l'equazione che permette di determinare il tempo di attivazione del monostabile in funzione dei componenti associati:

I primi componenti che si devono montare sullo stampato sono le resistenze e i condensatori non polarizzati

$$T = Cx \times Rx$$

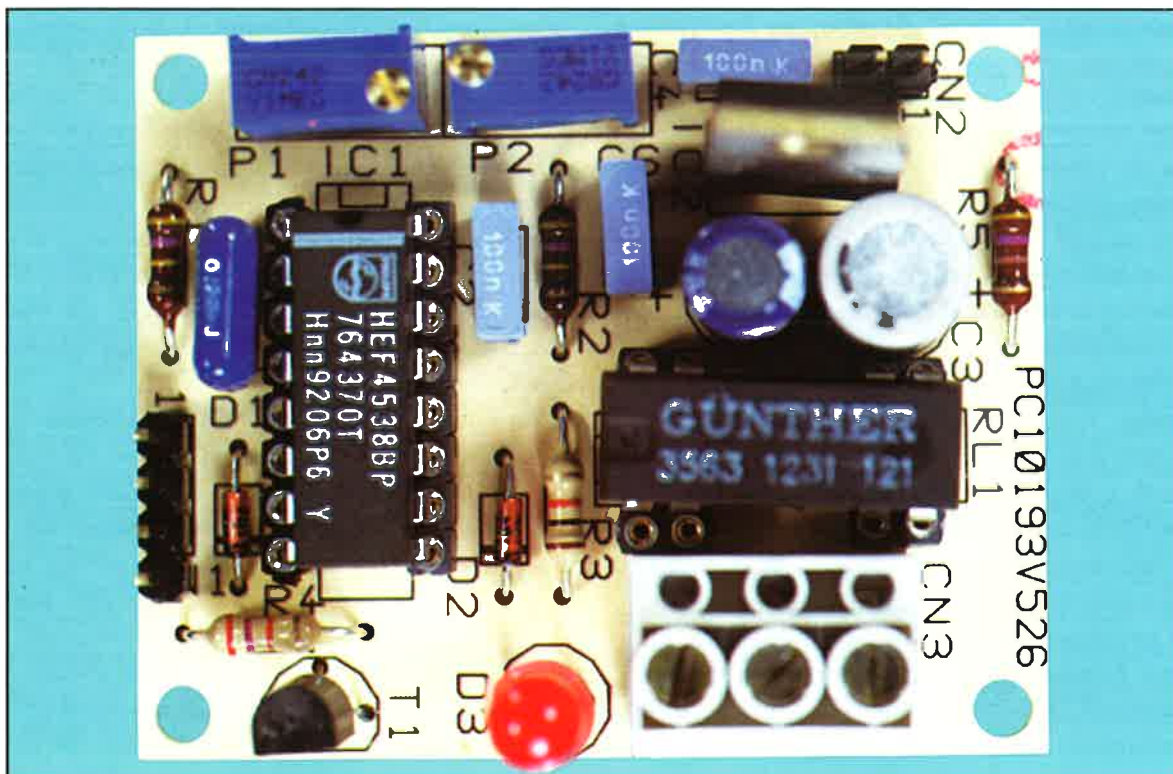
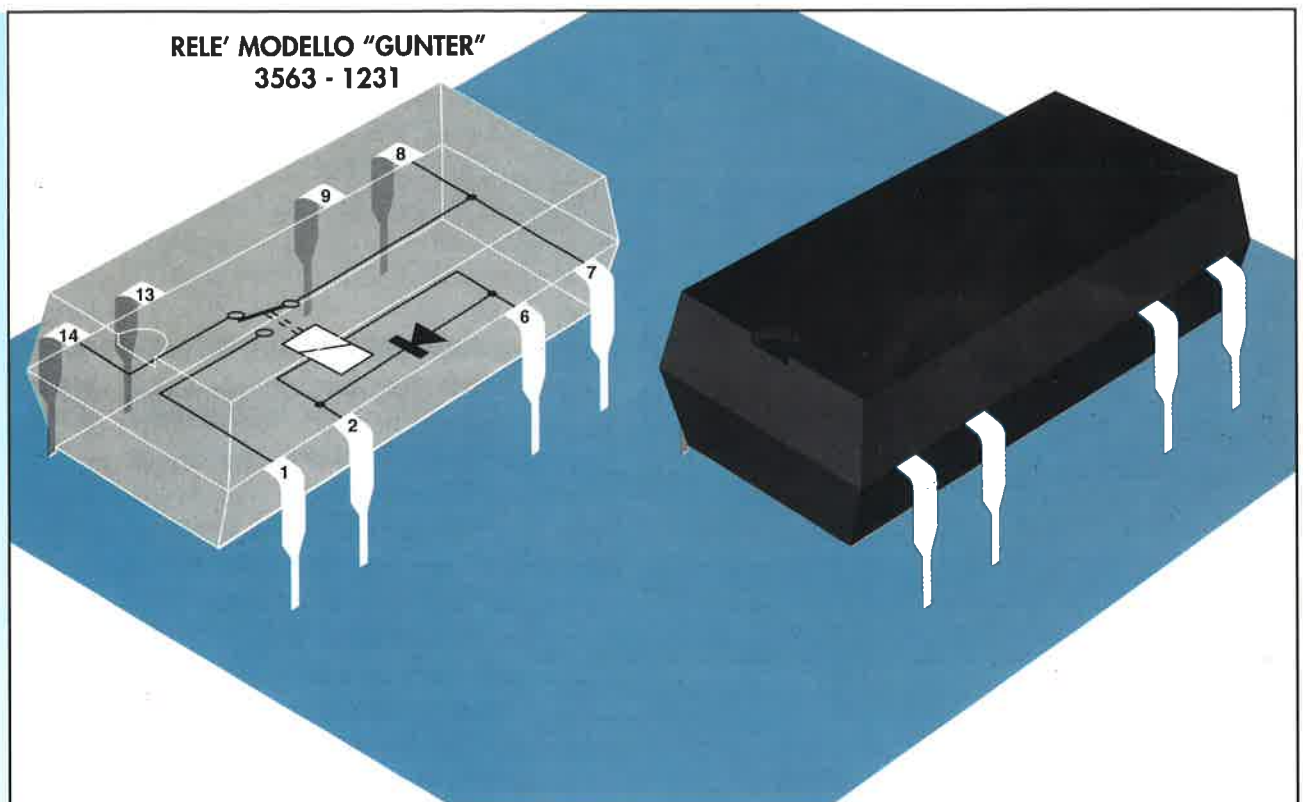
Dove T è il tempo (del punto o della linea) in secondi, Cx è la capacità (C1 o C2) espressa in Farad e Rx la resistenza (P1+R1 o P2+R2) espressa in Ohm.

Le uscite dei monostabili vengono collegate tra di loro tramite i diodi D1 e D2, e vanno a pilotare il transistor T1. Questo transistor è il componente che a sua volta pilota il relè di uscita, i cui contatti (aperti e chiusi) commutano alla frequenza del segnale Morse inviato dal computer.

Come si può osservare dallo schema, i segnali relativi ai simboli provengono dai terminali 1 e 2 del connettore CN1, mentre il terminale 3 ha una funzione ben specifica; poiché computer diversi (XT, AT, 386, ecc.) inviano i simboli all'interfaccia a velocità differenti a causa della loro diversa frequenza di lavoro, può accadere che i calcolatori più veloci risultino troppo rapidi, e inviino un simbolo senza attendere che il monostabile abbia terminato il ciclo precedente. Il segnale prelevato al terminale 3 aiuta a risolvere questo problema. La soluzione, in parte hardware e in parte software, è la seguente: quando si invia il primo simbolo (un punto o una linea) si attiva il monostabile corrispondente, l'uscita di questo raggiunge il livello

I condensatori elettrolitici, i terminali, e le prese devono essere montati in seguito

A questo punto si possono montare sullo stampato gli zoccoli per l'integrato e per il relè (facoltativo)

di innesco del transistor T1, e di conseguenza viene pilotato il relè.
Mentre il segnale è presente sulla base del transistor, il ritorno al computer (terminale 3 di CN1) si trova a livello logico alto. Il programma di gestione del circuito utilizza questa informazione per evitare che venga inviato all'interfaccia un nuovo simbolo prima che l'elaborazione del precedente sia terminata.
Il resto del circuito è formato da uno stadio di filtraggio e di stabilizzazione dell'alimentazione, e da un indicatore a LED che rivela lo stato attivo del relè. Giunti a questo punto non resta che darsi da fare e iniziare il montaggio.
Il segnale presente all'uscita dell'interfaccia non solo è controllato dal programma che lo gestisce, ma consente anche una regolazione di tipo hardware. Come si può osservare, i potenziometri P1 e P2 fanno parte della coppia resistiva incaricata di determinare il tempo di attivazione dell'integrato; la regolazione di questi due potenziometri permette di variare la lunghezza reale delle linee e dei punti.
Come già detto in precedenza, si è cercato di fornire un intervallo di lavoro opportuno, ma ciò non esclude che il lettore possa autoregolare questo range a suo piacimento. Una piccola routine software che attivi sequenzialmente i monostabili, e l'aiuto visivo del diodo LED sono tutto ciò che serve per portare a termine questa regolazione.

La regolazione dei potenziometri consente di impostare la variazione della lunghezza reale delle linee e dei punti

Aspetto della scheda completamente montata

Il relè utilizzato è esternamente simile ad un integrato. Nella figura si possono osservare i suoi collegamenti interni

MONTAGGIO DELLA SCHEDA

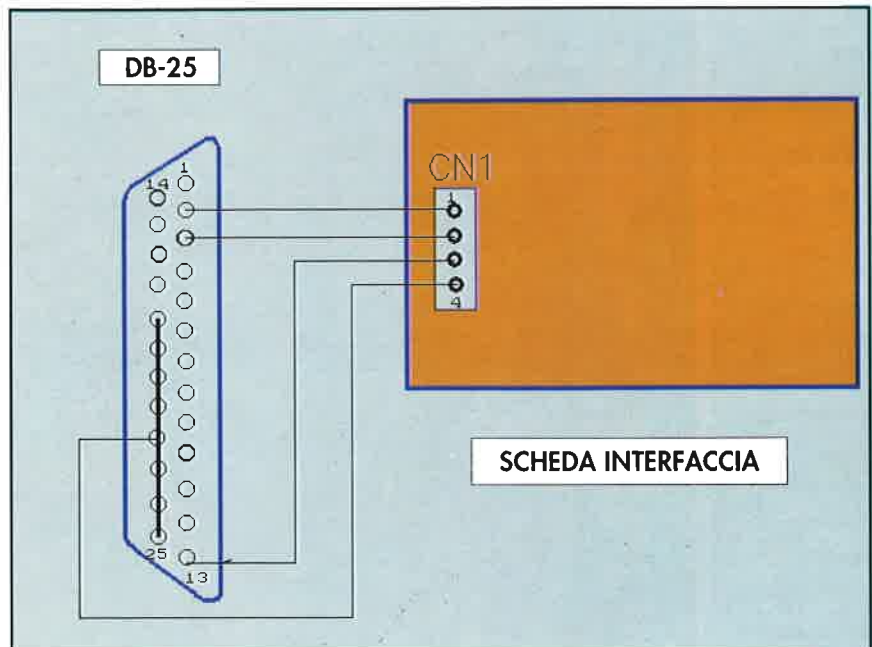
Dopo che si sono acquistati e identificati tutti i componenti necessari, il montaggio dovrà essere eseguito seguendo uno schema logico. Il primo passo consiste nell'inserimento e nella saldatura dei componenti passivi, costituiti dalle resistenze e dai condensatori non polarizzati. Si prosegue con i condensatori elettrolitici, che dovranno essere montati sullo stampato rispettando rigorosamente la polarità indicata; il riferimento relativo al terminale negativo di questi componenti è stampato sul contenitore, mentre il riferimento relativo al terminale positivo è serigrafato sul circuito stampato.
Dopo aver inserito le resistenze e i condensatori si può proseguire con i terminali dei connettori CN1 e CN2, con il connettore a vite a tre posizioni CN3, e con i terminali torniti femmina che servono da zoccoli per gli integrati.

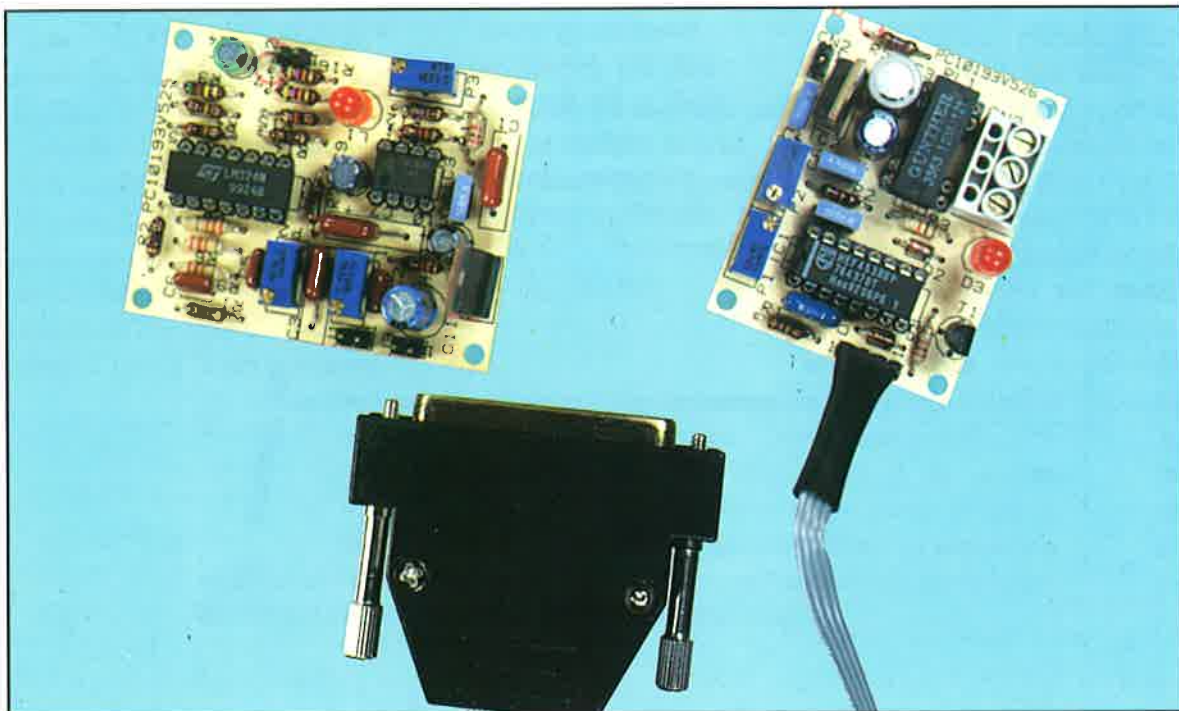
Gli ultimi componenti che si devono montare sono gli integrati e i semiconduttori

L'ultima operazione è quella relativa al montaggio dei semiconduttori e all'inserimento dei circuiti integrati. Si inizia saldando IC2 (la cui corretta posizione è indicata sulla serigrafia del circuito stampato), per proseguire con i diodi D1 e D2 (il cui catodo è identificato da una striscia di colore nero stampata sul contenitore del componente); infine si può inserire il transistor T1, la cui corretta orientazione è riportata anch'essa dalla serigrafia presente sul circuito stampato. Il diodo LED deve essere inserito facendo coincidere il profilo del contenitore con la forma disegnata sullo stampato, mentre l'integrato IC1 deve essere inserito nel relativo zoccolo con l'orientamento definito dalla tacca di riferimento riportata dalla serigrafia; lo stesso discorso vale anche per il relè, sia che venga montato su zoccolo sia che venga saldato direttamente sullo stampato.
Nella figura corrispondente si può osservare lo schema interno del relè utilizzato. Viene consigliato il suo impiego per le piccole dimensioni e per la praticità di montaggio, in quanto esternamente è identico ad un circuito integrato. Tuttavia, in caso di sostituzione, lo schema allegato può essere molto utile per facilitare una nuova scelta. Si fa notare che nel caso sia necessario sostituire il transistor indicato nell'elenco componenti con uno equivalente (nell'ipotesi che il primo non sia

reperibile), bisognerà verificare che la disposizione dei suoi terminali coincida con quella del BC546B, che sia un NPN, e che le caratteristiche siano all'incirca le stesse.
Si ricorda che l'alimentazione del circuito viene prelevata dall'uscita del regolatore integrato IC2 (+5 Vcc), mentre il relè viene alimentato a +12 Vcc.

Collegamenti con il PC

Come nel caso del circuito decodificatore presentato nel capitolo precedente, anche il circuito codificatore Morse viene collegato alla porta parallela (Centronics) del personal computer, che risulta sufficientemente versatile. Probabilmente il lettore conosce la potenzialità della porta Centronics solo per gli impieghi legati al controllo della propria stampante, ma è ovvio che le possibilità offerte da questo connettore sono molteplici, grazie alla notevole quantità di segnali presenti sui suoi terminali e alla possibilità di trasmissione bidirezionale.
Di seguito viene riportata, a titolo puramente indicativo, la distribuzione del pin-out di questa porta: sui 25 terminali del classico connettore DB-25 sono presenti quattro gruppi principali di segnali. Il primo di questi comprende le linee dei dati D0-D7 (terminali dal 2 al 9); il secondo gruppo viene definito porta di controllo, ed è

I terminali da saldare sul connettore DB-25 sono quelli riportati in questo schema

Il cavo di collegamento al PC deve essere inserito nel connettore CN1 presente sulla scheda di codifica

Il circuito di codifica Morse viene collegato alla porta parallela (Centronics) del personal computer

Elenco componenti

Resistenze

R1, R2 = 470 kΩ

R3 = 10 kΩ

R4 = 2,7 kΩ

R5 = 470 Ω

P1, P2 = 1 MΩ potenziometri

Condensatori

C1 = 300 nF, ceramico

C2, C4, C6 = 100 nF, ceramico

C3 = 100 μF, elettrolitico

C5 = 47 μF, elettrolitico

Semiconduttori

IC1 = CD4538, monostabile

IC2 = LM7805, regolatore di tensione a 5 V

D1, D2 = 1N4148

D3 = Diodo LED rosso,

T1 = BC546B, transistor NPN

Varie

RL1, relè reed a 12 V (Gunther 3563-1231)

CN1 = terminali maschi (4x)

CN2 = terminali maschi (2x)

CN3 = connettore a vite tre vie per c.s.

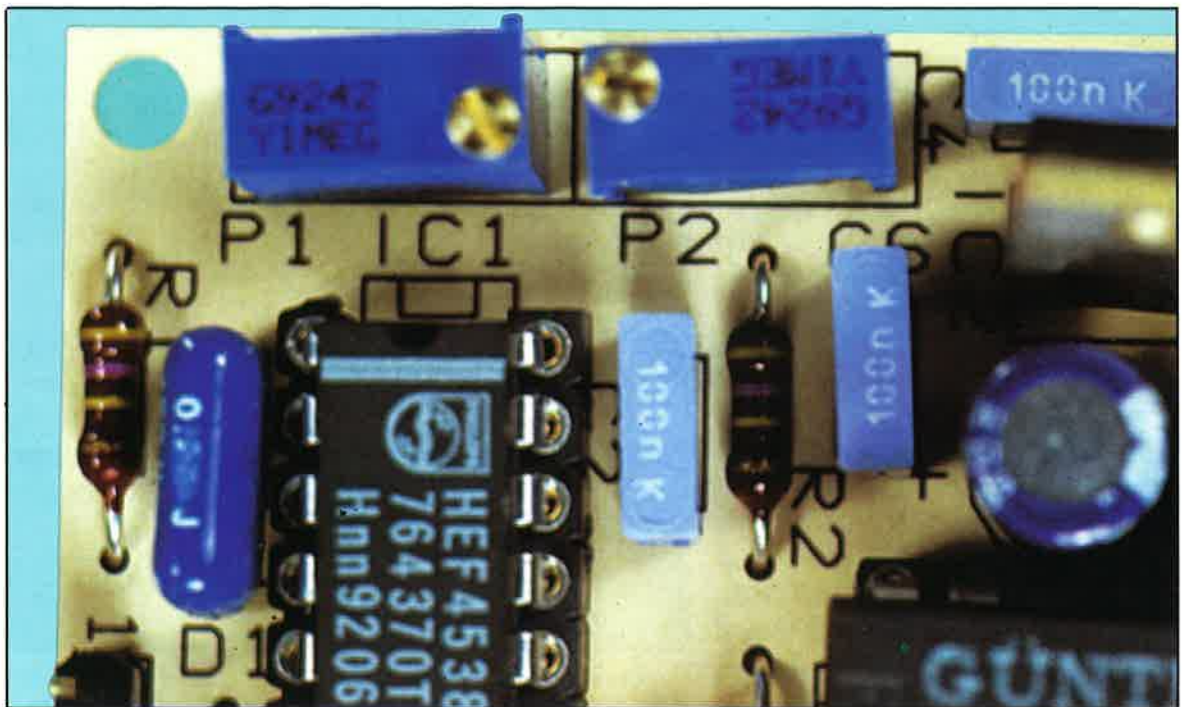
Connettore DB-25 maschio

Connettore portapile

Terminali femmina per il collegamento a CN1 (4x)

Terminali femmina per collegamento del connettore portapile a CN2

Cavo piatto a 4 fili

Circuito stampato PC10193V526.

I potenziometri P1 e P2 permettono di regolare il tempo di attivazione dei monostabili

composto dai terminali 1, 14, 16 e 17. Il terzo gruppo corrisponde alla porta di stato (o più semplicemente "status"), mentre l'ultimo gruppo raccoglie le masse di sistema (terminali dal 18 al 25).

In pratica, i segnali che arrivano all'interfaccia sono D0 e D1 (dalla porta dei dati), e S4 (dalla porta di stato), che corrispondono rispettivamente ai terminali 2, 3 e 13 del DB-25. I terminali dal 18 al 25 sono collegati a massa. Per collegare il circuito al PC, i lettori che hanno realizzato il montaggio della scheda per la decodifica del codice Morse possono utilizzare il cavo di collegamento già costruito per questo dispositivo. Nella figura corrispondente è comunque riportato lo schema per eseguirne uno dedicato. L'ultimo problema che bisogna affrontare, se si desiderano collegare entrambi i circuiti contemporaneamente, è costituito dall'alimentazione. Poiché ambedue i circuiti sono dotati di regolatore integrato di tensione, è possibile alimentarli contemporaneamente con lo stesso alimentatore. Per i lettori più curiosi si riporta anche un esempio di trasmissione in codice Morse, anche se il livello di automazione che implica il montaggio presentato rende questa informazione puramente indicativa.

I segnali che arrivano alla scheda di interfaccia sono D0, D1 (porta dei dati) e S4 (porta di stato), e corrispondono rispettivamente ai terminali 2, 3 e 13 del DB-25